

1/28

Fig. 1

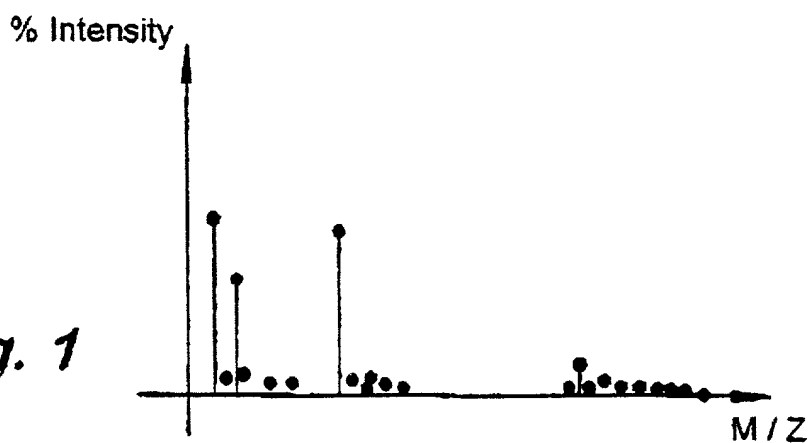


Fig. 2

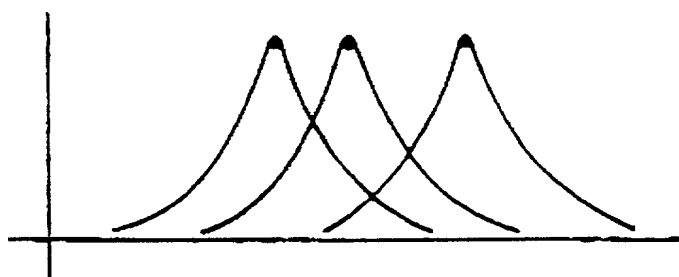


Fig. 3

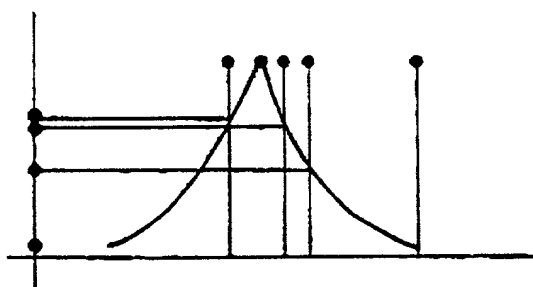
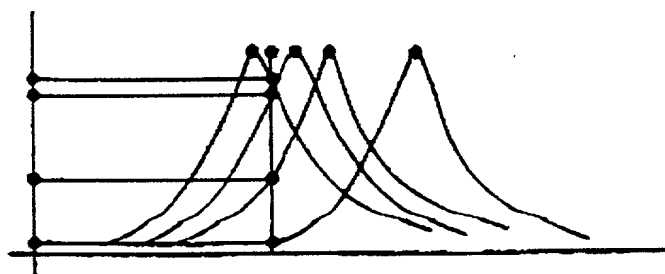


Fig. 4



2/28

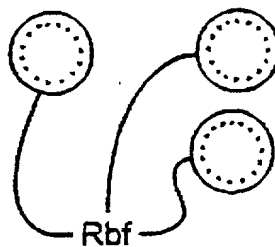
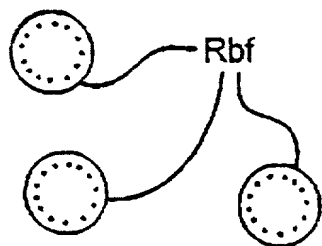
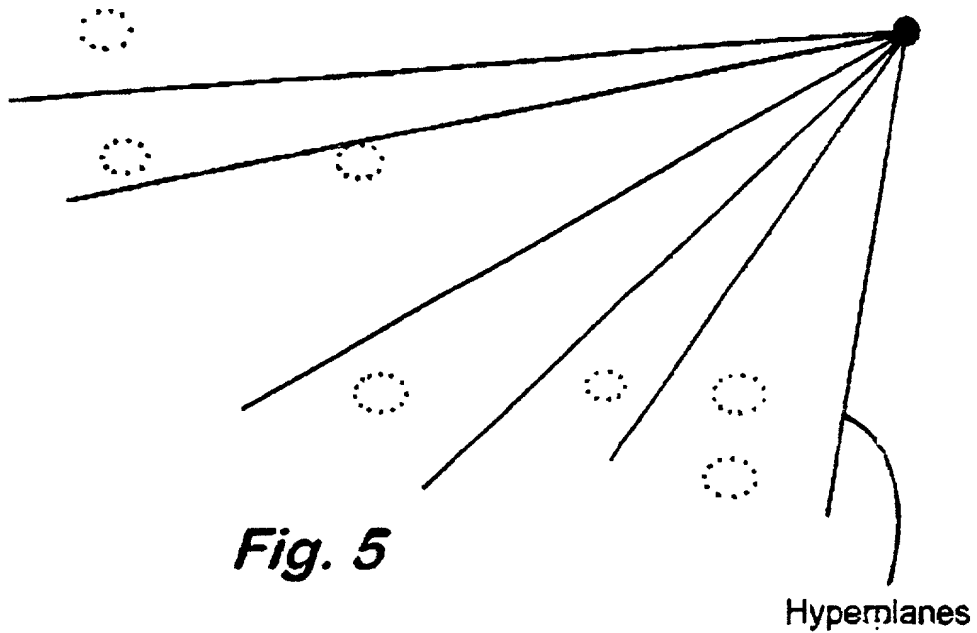
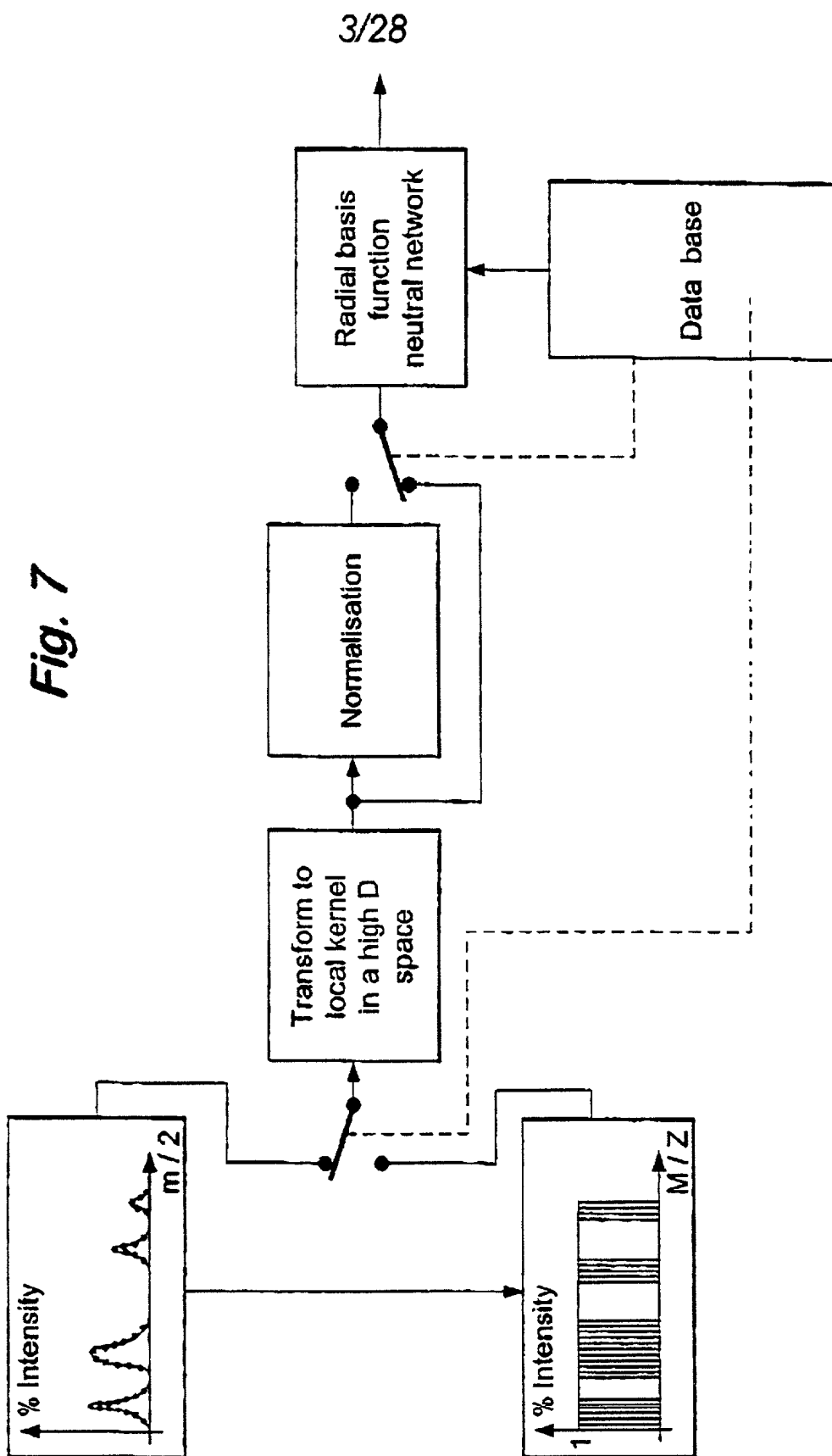
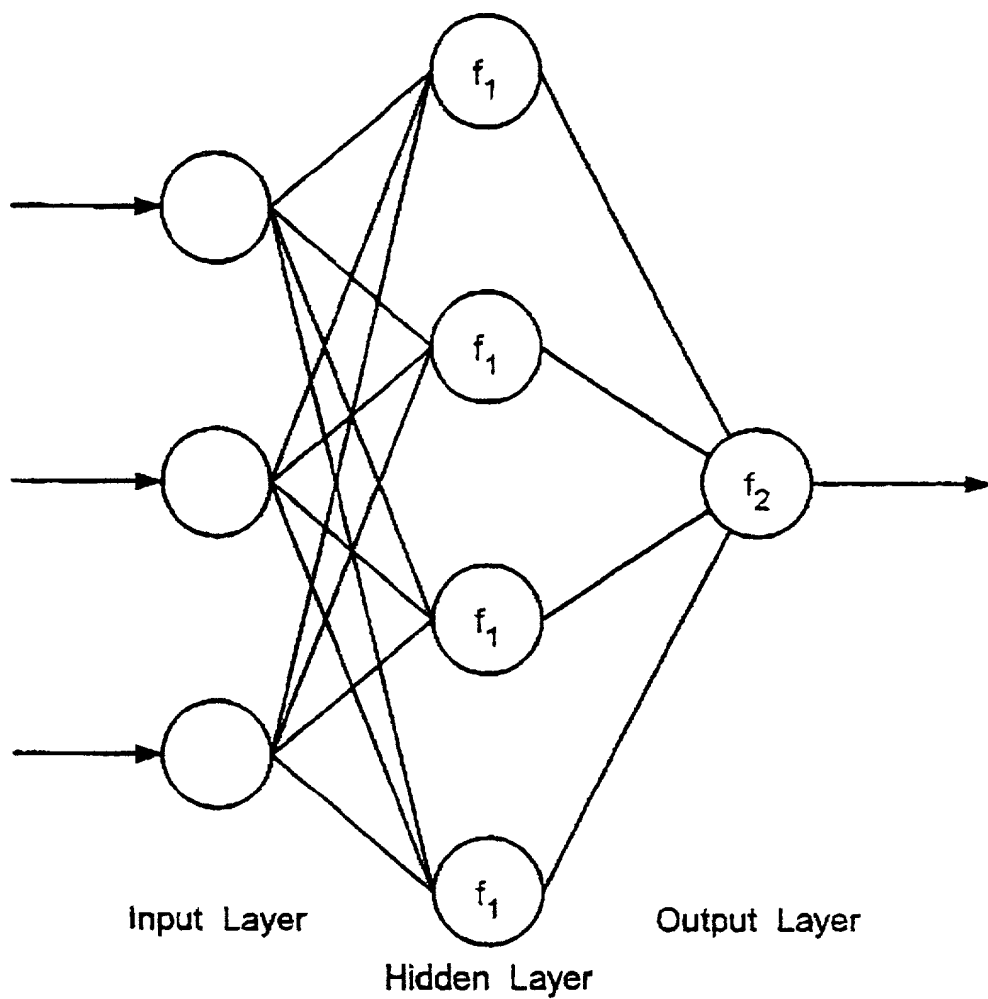


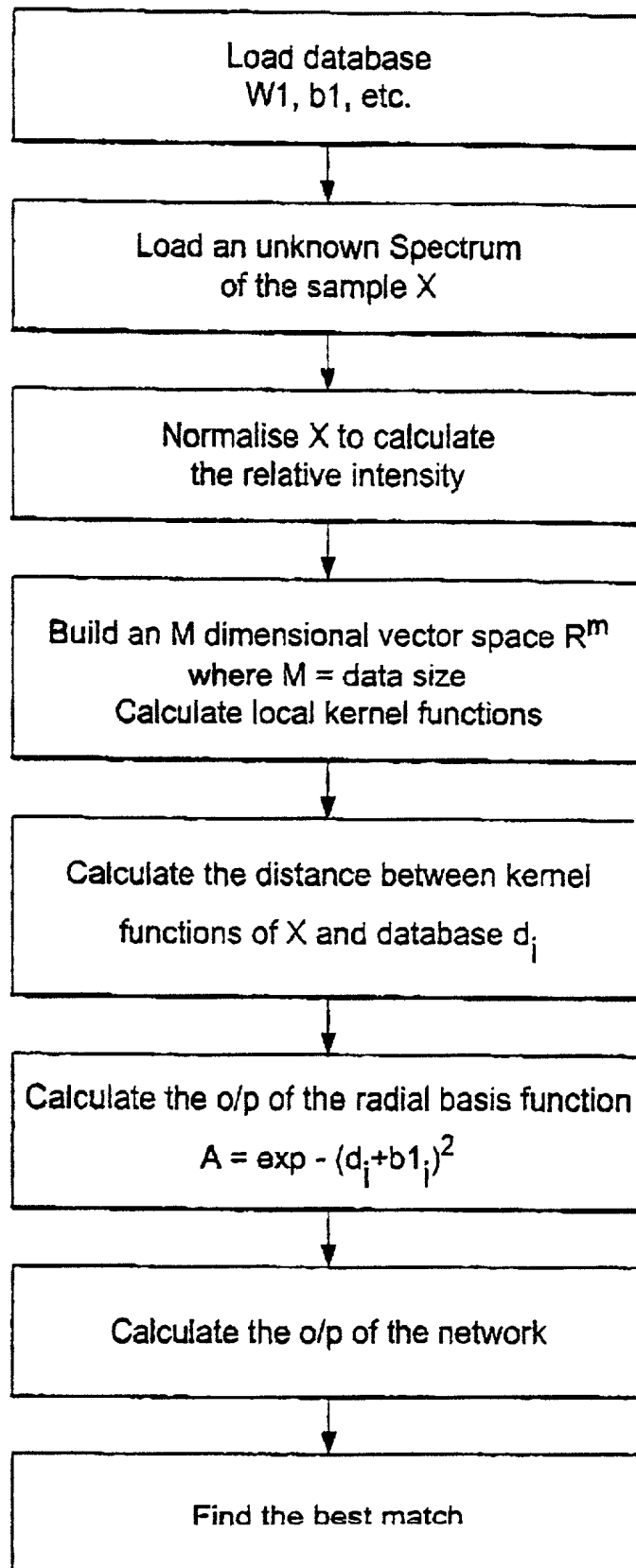
Fig. 6



4/28

**Fig. 8**

5/28

Fig. 9

6/28

Fig. 10a

Program Listing

```
#include <formatio.h>
#include <analysis.h>

#include <utility.h>
#include <ansi_c.h>
#include <userint.h>
#include "sample4.h"

#define NoData      125
#define DataSize    16370

static int panelHandle ;

    static int status ;
    static FILE *file_handle ;
    static char pathname [ MAX_PATHNAME_LEN ] ;
    static char directory [ MAX_PATHNAME_LEN ] ;

int Load_Sample_ok=0 ;
int Load_DataBase_OK=0 ;

int err ;
/*double mean_value ;
double datapoints [ 100 ] ;
*/

double P [ DataSize ] ;
double P_index [ DataSize ] ;

double w1 [ NoData ] [ DataSize ] ;
double w2 [ NoData ] [ NoData ] ;
double b1 [ NoData ] ;
double b2 [ NoData ] ;

double a1 [ NoData ] ;
double a2 [ NoData ] ;
```

7/28

Fig. 10b

```

double n [ NoData ];
double x [ DataSize ];
double y [ DataSize ];
double y2 [ DataSize ];
double dist [ NoData ];

```

```

/*****

```

```

int main ( int argc, char *argv [ ] )
{
    if ( InitCVRTE ( 0, argv, 0 ) == 0 ) /* Needed if linking in external
    compiler; harmless otherwise */
        return -1; /* out of memory */
    if ( ( panelHandle = LoadPanel ( 0, "sample4.uir", PANEL ) ), < 0 )
        return -1;

    DisplayPanel ( panelHandle );
    RunuserInterface ( );
    return 0 ;
}

```

```

/*****

```

```

int CVICALLBACK Shutdown ( int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2 )
{
    switch ( event )
    {
        case EVENT_COMMIT:
            QuitUserInterface ( 0 );
            break ;
        case EVENT_RIGHT_DOUBLE_CLICK:
            break ;
    }
    return 0 ;
}

```

```

/*****

```

8/28

Fig. 10c

```

int CVICALLBACK Shutdown ( int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2 )
{
    double norm;

    int i=0;
    int j;
    double M, m;
    char buff[80];

    int err, s, q;

    switch ( event ) {

        case EVENT_COMMIT:

            if ( Load_DataBase_OK )
            {
                if (Load_Sample_OK)
                {

                    DeleteGraphPlot ( panelHandle, PANEL_WAVEFORM_2, -1
                                     VAL_IMMEDIATE_DRAW );

                    SetCtrlVal ( panelHandle, PANEL_ELEMENT, " Searching,
                                                                    Please Wait " );

                    for ( s=0; s<NoData; s++ )
                    {
                        for ( q=0; q<DataSize; q++ )
                        {
                            x[q] = w1[s][q];
                            /* x[q] = ( x[q]) * ( x[q] - P[q] );
                               dist[s] = dist[s] + x[q];
                            */
                        }
                    }
                }
            }
        }
    }

```


9/28

Fig. 10d

```

norm=0;
Sub1D ( x, P, DataSize, y );
Mul1D ( y, y, DataSize, Y2 );
Sum1d ( y2, DataSize, &norm );
norm=sqrt ( norm );
dist [ s ] = norm ;

/*      dist [ s ] = sqrt ( dist [ s ] ); */
n [ s ] = dist [ s ] * b1 [ s ];
a1 [ s ] = exp ( -n [ s ] * n [ s ] );

/*
for ( q=0 ; q<NoData; q++ )
{

a1 [ s ] = w2 [ s ] * a1 [ s ] + b2 [ s ];
n [ i ] = n [ i ] * b;
a1 [ i ] = exp ( -n [ i ] * n [ i ] );
datapoints [ i ] = rand ( ) / 32768.0 ;
}

}

MaxMin1D ( a1, NoData, &M, &i, &m, &j );
QScale1D ( a1, NoData, a1, &M );
j=i+1;

PlotY ( panelHandle, PANEL_WAVEFORM_2, a1, NoData,
VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE,
VAL_SOLID, 1, VAL_WHITE );

switch ( j )
{

case 0:

Fmt ( buff, "%s ( %d )", "No Sample was selected", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

break :

```

10/28

Fig. 10e

case 1:

```

Fmt ( buff, "%s (%d)", " Acina, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 2:

```

Fmt ( buff, "%s (%d)", " alcal, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 3:

```

Fmt ( buff, "%s (%d)", " baccar, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 4:

```

Fmt ( buff, "%s (%d)", " baccar2, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 5:

```

Fmt ( buff, "%s (%d)", " bacmy, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 6:

```

Fmt ( buff, "%s (%d)", " bacsub, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

11/28

Fig. 10f

break ;

case 9:

```

Fmt ( buff, " %s ( %d ) ", " citd, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 8:

```

Fmt ( buff, " %s ( %d ) ", " citd2, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 9:

```

Fmt ( buff, " %s ( %d ) ", " citf, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 10:

```

Fmt ( buff, " %s ( %d ) ", " citf2, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 11:

```

Fmt ( buff, " %s ( %d ) ", " citf3, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 12:

12/28

Fig. 10g

```
Fmt ( buff, " %s ( %d ) ", " ecoli1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 13:
```

```
Fmt ( buff, " %s ( %d ) ", " ecoli2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 14:
```

```
Fmt ( buff, " %s ( %d ) ", " ecoli3, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 15:
```

```
Fmt ( buff, " %s ( %d ) ", " ecoli4, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 16:
```

```
Fmt ( buff, " %s ( %d ) ", " ecoli26, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 17:
```

```
Fmt ( buff, " %s ( %d ) ", " ecoli27, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

13/28

Fig. 10h

case 18:

```
Fmt ( buff, " %s ( %d ) ", " ecoli28, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 19:

```
Fmt ( buff, " %s ( %d ) ", " ecoli29, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 20:

```
Fmt ( buff, " %s ( %d ) ", " ecoli30, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 21:

```
Fmt ( buff, " %s ( %d ) ", " ecoli31, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 22:

```
Fmt ( buff, " %s ( %d ) ", " ecoli32, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 23:

```
Fmt ( buff, " %s ( %d ) ", " ecoli33, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

SUBSTITUTE SHEET (RULE 26)

T02050" 68924860

14/28

Fig. 10i

break ;

case 24:

```

Fmt ( buff, " %s ( %d ) ", " ecoli34, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 25:

```

Fmt ( buff, " %s ( %d ) ", " ecoli35, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 26:

```

Fmt ( buff, " %s ( %d ) ", " ent, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 27:

```

Fmt ( buff, " %s ( %d ) ", " entc, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 28:

```

Fmt ( buff, " %s ( %d ) ", " entd, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
break ;

```

case 29:

```
Fmt ( buff, " %s ( %d ) ", "entf. Element No.", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
Fmt ( buff, " %s ( %d ) ", "entf2, Element No.", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
Fmt ( buff, "%s (%d)", "ent3, Element No.", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```

Fmt ( buff, "%s (%d)", "entf4, Element No.", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```
break ;
```

```
Fmt ( buff, "%s (%d)", "entf5, Element No.", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```

Fmt ( buff, " %s ( %d ) ", "entf6, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```
break ;
```

16/28

Fig. 10k

case 35:

```
Fmt ( buff, " %s ( %d ) ", " entf7, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 36:

```
Fmt ( buff, " %s ( %d ) ", " entvre, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 37:

```
Fmt ( buff, " %s ( %d ) ", " gon1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 38:

```
Fmt ( buff, " %s ( %d ) ", " gon2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 39:

```
Fmt ( buff, " %s ( %d ) ", " hafalv, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 40:

```
Fmt ( buff, " %s ( %d ) ", " keba, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```


17/28

Fig. 101

break ;

case 41:

```

Fmt ( buff, " %s ( %d ) ", " listg, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 42:

```

Fmt ( buff, " %s ( %d ) ", " listi, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 43:

```

Fmt ( buff, " %s ( %d ) ", " listm, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 44:

```

Fmt ( buff, " %s ( %d ) ", " listm3, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 45:

```

Fmt ( buff, " %s ( %d ) ", " listm4, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
break ;

```

case 46:

18/28

Fig. 10m

```

Fmt ( buff, " %s ( %d ) ", " listmu, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 47:

```

```

Fmt ( buff, " %s ( %d ) ", " mening1, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 48:

```

```

Fmt ( buff, " %s ( %d ) ", " mening2, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 49:

```

```

Fmt ( buff, " %s ( %d ) ", " nelong, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 50:

```

```

Fmt ( buff, " %s ( %d ) ", " nflav1, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 51:

```

```

Fmt ( buff, " %s ( %d ) ", " nflav2, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

19/28

Fig. 10n

case 52:

```

Fmt ( buff, " %s ( %d ) ", " nsicca, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 53:

```

Fmt ( buff, " %s ( %d ) ", " pro1, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 54:

```

Fmt ( buff, " %s ( %d ) ", " pro2, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 55:

```

Fmt ( buff, " %s ( %d ) ", " pro3, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 56:

```

Fmt ( buff, " %s ( %d ) ", " prov1, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

break ;

case 57:

```

Fmt ( buff, " %s ( %d ) ", " pseu1, Element No. " , i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

20/28

Fig. 10o

```
break ;
```

```
case 58:
```

```
Fmt ( buff, " %s ( %d ) ", "pseua2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 59:
```

```
Fmt ( buff, " %s ( %d ) ", "pseua3, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 60:
```

```
Fmt ( buff, " %s ( %d ) ", "sal, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 61:
```

```
Fmt ( buff, " %s ( %d ) ", "salg1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 62:
```

```
Fmt ( buff, " %s ( %d ) ", "salg10, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );  
break ;
```

```
case 63:
```

21/28

Fig. 10p

```

Fmt ( buff, "%s (%d)", "salg78, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 64:

```

```

Fmt ( buff, "%s (%d)", "salt2, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 65:

```

```

Fmt ( buff, "%s (%d)", "saltyp, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 66:

```

```

Fmt ( buff, "%s (%d)", "saltyp1, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 67:

```

```

Fmt ( buff, "%s (%d)", "serratt, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

case 68:

```

```

Fmt ( buff, "%s (%d)", "shig, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

22/28

Fig. 10q

case 69:

```
Fmt ( buff, " %s ( %d ) ", " stapha1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 70:

```
Fmt ( buff, " %s ( %d ) ", " stapha2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 71:

```
Fmt ( buff, " %s ( %d ) ", " stapha3, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 72:

```
Fmt ( buff, " %s ( %d ) ", " strep1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 73:

```
Fmt ( buff, " %s ( %d ) ", " strep2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

case 74:

```
Fmt ( buff, " %s ( %d ) ", " beco16k1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

23/28

Fig. 10r

```
break ;
```

```
case 75:
```

```
Fmt ( buff, " %s ( %d ) ", " beco16k2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 76:
```

```
Fmt ( buff, " %s ( %d ) ", " beco16k3, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 77:
```

```
Fmt ( buff, " %s ( %d ) ", " beco16k4, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 78:
```

```
Fmt ( buff, " %s ( %d ) ", " psta16k1, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );
```

```
break ;
```

```
case 79:
```

```
Fmt ( buff, " %s ( %d ) ", " psta16k2, Element No. " , i );  
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );  
break ;
```

```
case 80:
```

24/28

Fig. 10s

```

Fmt ( buff, "%s (%d)", "psta16k3, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

default:

```

```

Fmt ( buff, "%s (%d)", "Unknown, Element No. ", i );
SetCtrlVal ( panelHandle, PANEL_ELEMENT, buff );

```

```

break ;

```

```

}

```

```

YGraphPopup ( " Result of Search Engine for the Selected Sample ",
               a1, NoData, VAL_DOUBLE );

```

```

}

```

```

else
{

```

```

MessagePopup ( " SAMPLE ", " Please Load a Sample First " );

```

```

}

```

```

}

```

```

else
{

```

```

MessaagePopup ( " DATA BASE ", " Please Load the Data Base First " );

```

```

}

```

```

break ;

```

```

case EVENT_RIGHT_DOUBLE_CLICK:

```

```

}

```


25/28

Fig. 10t

```

    return 0 ;
}

/*****

int CVICALLBACK LoadDataBase ( int panel, int control, int event,
                               void *callbackData, int eventData1, int eventData2 )
{
    int s, q, num ;

    switch ( event )
    {
        case EVENT_COMMIT:

            SetCtrlVal ( panelHandle, PANEL_ELEMENT, " Loading, Please Wait " ) ;

            GetProjectDir ( pathname ) ;
            status = FileSelectPopup ( directory, " w100.mat ", "*.Mat ( Mat file ) ",
                                     " Data File ", VAL_LOAD_BUTTON, 0, 0, 1, 0,
                                     pathname ) ;

            if ( status != VAL_NO_FILE_SELECTED )
            {
                file_handle = fopen ( pathname, " r " ) ;
                for ( s=0 ; s<NoData ; s++ )
                {
                    for ( q=0 ; q<DataSize ; q++ )
                    {
                        num = fscanf ( file_handle, " %1f ", &w1 [ s ] [ q ] ) ;
                    }
                }
                fclose ( file_handle ) ;
                Load_DataBase_OK=1 ;
            }
            else
            {

```

26/28

Fig. 10u

```

Load_DataBase_OK=0;

}

GetProjectDir ( pathname );
status = FileSelectPopup ( directory, " b100.mat ", " *.Mat ( Mat file ) ",
                          " Data File ", VAL_LOAD_BUTTON, 0, 0, 1, 0,
                          pathname );
if ( status != VAL_NO_FILE_SELECTED )
{
    file_handle = fopen ( pathname, " r " );
    for ( s=0 ; s<NoData ; s++ )
    {
        num = fscanf ( file_handle, " %1f ", &b1 [ s ] );
    }
    fclose ( file_handle );
}
else
{
    Load_DataBase_ok=0;
}

SetCtrlVal ( panelHandle, PANEL_ELEMENT, " Load a sample or search " );

    break ;

    case EVENT_RIGHT_DOUBLE_CLICK:

        break ;

}
return 0 ;
}
/*****
int CVICALLBACK Load_Sample ( int panel, int control, int event,
SUBSTITUTE SHEET (RULE 26)

```

68524860

27/28

Fig. 10v

```

void *callbackData, int eventData1, int eventData2 )
{
    int i, num, pmax_index, pmin_index;
    double pmax, pmin;
    switch ( event )
    {

        case EVENT_COMMIT:

            DeletGraphPlot ( panelHandle, PANEL_WAVEFORM, -1,
                               VAL_IMMEDIATE_DRAW );

            SetCtrlVal ( panelHandle, PANEL_ELEMENT, " Loading, Please Wait " );

            GetProjectDir ( pathname );
            status = FileSelectPopup ( directory, "*.16k ", "*.dat ( data.dat )",
                                     " Data File ", VAL_LOAD_BUTTON, 0, 0, 1, 0,
                                     pathname );

            if ( status != VAL_NO_FILE_SELECTED )
            {
                file_handle = fopen ( pathname, " r " );

                for ( i=0 ; i<DataSize ; i++ )

                {

                    num = fscanf ( file_handle, " %1f %1f \n ", &P_index [i], &P [i] );

                }

                fclose ( file_handle );

                Load_Sample_OK=1 ;

                SetCtrlVal ( panelHandle, PANEL_ELEMENT, " Unknown Sample " );

                MaxMin1D ( P, DataSize, &pmax, &pmax_index, &pmin, &pmin_index );

```

28/28

Fig. 10w

```

QScale1D ( P, DataSize, P_index, &pmax );

for ( i=0 ; i<DataSize ; i++ ) P [ i ] = 3000 * P_index [ i ];

PlotY ( panelHandle, PANEL_WAVEFORM, P_index, DataSize,
        VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE,
        VAL_SOLID, 1, VAL_YELLOW );

/* PlotY ( panelHandle, PANEL_WAVEFORM, P, DataSize, VAL_DOUBLE,
        VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1,
        VAL_YELLOW );

/* SetAxisScalingMode ( panelHandle, PANEL_WAVEFORM, VAL_XAXIS,
        VAL_MANUAL, 500, 10000 );
/* SetAxisRange ( panelHandle, PANEL_WAVEFORM, VAL_NO_CHANGE,
        500, 10000, VAL_AUTOSCALE, 0.0, 1.0 );
*/

}
else
{
Load_Sample_OK=0;

}

break;

case EVENT_RIGHT_DOUBLE_CLICK:

break;

}

return 0;
}

```
